

Plugin pro aplikaci Autodesk Revit Architecture

Autodesk Revit Architecture Plugin

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava*.

V Ostravě 6. května 2011

.....

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 6. května 2011

.....

Rád bych poděkoval svému vedoucímu bakalářské práce, panu Ing. Michalu Krumníkovi za pomoc při její tvorbě, dále Ing. Jiřímu Drahotuskému za testování programové části této práce z pohledu dlouholetého projektanta a uživatele aplikace Revit Architecture a nakonec společnosti Adeon CZ, s.r.o. za poskytnutí potřebné licence programu a za projekty a data, jež byly využity při testování nadstavby.

Abstrakt

Cílem této práce je tvorba nadstavbové aplikace pro software Autodesk Revit Architecture 2011, která umožňuje automatizovanou tvorbu dynamických legend popisujících modelové objekty projektu. Práce se ve své teoretické části věnuje popisu technologie BIM, jejím specifickým oproti ostatním metodám počítačem asistovaného kreslení a aplikacím, jež ji využívají. Zvláštní důraz je kladen na program Revit Architecture. V praktické části je popsáno a zdokumentováno aplikační programovací rozhraní Revitu a jsou představeny algoritmy vytvořené nadstavby.

Klíčová slova: Revit Architecture, BIM, CAD, Revit API, plugin, legendy

Abstract

The goal of this thesis is creation of an add-on application for Autodesk Revit Architecture 2011 which would enable automated creation of dynamic legends describing model elements in a Revit project. In its theoretic part, the thesis describes the technology called BIM, it's specifics compared to other methods of computer assisted drawing and applications that have been using it. Special emphasis has been put onto Revit Architecture. The practical part of the thesis is devoted to introducing and explaining Revit application programming interface and some of the algorithms used in the created add-on application.

Keywords: Revit Architecture, BIM, CAD, Revit API, plugin, legends

Seznam použitých zkratk a symbolů

ADT	– Architectural Desktop
API	– Application Programming Interface
BIM	– Building Information Modeling
CAD	– Computer-Aided Design
CAE	– Computer-Aided Engineering
CAM	– Computer-Aided Manufacturing
CNC	– Computer Numerical Controlled
GUI	– Graphical User Interface
GUID	– Globally Unique Identifier
IFC	– Industry Foundation Classes
LINQ	– Language Integrated Query
MEP	– Mechanical / Electrical / Plumbing
SDK	– Software Development Kit
TZB	– Technické Zařízení Budov
XML	– Extensible Markup Language

Obsah

1	Úvod	5
2	Informační model budovy	7
2.1	BIM a CAD software	7
2.2	Definice informačního modelu budovy	9
2.3	BIM aplikace	9
2.3.1	Graphisoft ArchiCAD	10
2.3.2	Bentley	10
2.3.3	Autodesk Revit	11
2.3.4	Ostatní	12
3	Autodesk Revit Architecture	13
3.1	Organizace dat	13
3.2	Podpora nadstaveb	13
3.3	Revit Architecture API	14
3.4	Umístění nadstaveb do prostředí Revitu	15
3.5	Hello World s využitím Revit API	15
3.5.1	Tvorba nového projektu	15
3.5.2	Třída HelloWorld	16
3.5.3	Začlenění do Revitu	17
4	Revit Legendy+	19
4.1	Struktura pluginu	19
4.1.1	Instalace a distribuce	20
4.1.2	Programové jádro	20
4.2	Algoritmus tvorby legendy	21
4.3	Legendy+ v prostředí Revitu	23
4.3.1	Připojit atribut	24
4.3.2	Zkontrolovat text	25
4.3.3	Odpojit atribut	26
4.3.4	Vytvořit legendu	26
4.4	Testování	29
5	Závěr	31
6	Literatura	32

Seznam tabulek

1	BIM software Graphisoftu	11
2	BIM software Autodesku	12
3	XML struktura souboru HelloWorld.addin	18

Seznam obrázků

1	Propojení profesí v rámci informačního modelu budovy	10
2	Struktura Revit API	14
3	Hello World v prostředí Revitu	18
4	Výkaz generovaný základními funkcemi Revitu	19
5	Projektová struktura Legendy+	20
6	Ribbon Legendy+ v panelu doplňků	24
7	Příkaz Připojit atribut	24
8	Okno příkazu Zkontrolovat text	26
9	Příkaz Vyvořit legendu	27
10	Legenda vygenerovaná nadstavbou	28

Seznam výpisů zdrojového kódu

1	Jmenné prostory	16
2	Třída HelloWorld	17
3	HelloWorld manifest	17
4	Zjednodušená podoba algoritmu tvorby tabulky rozkopírováním vzoru .	23
5	Algoritmus tvorby a připojení sdílených parametrů	25

1 Úvod

Je tomu již téměř půl století od doby, kdy byl představen historicky první software pro počítačem asistované kreslení (CAD), Sketchpad, který vznikl jako součást disertační práce Ivana Sutherlanda na Massachusettském Institutu Technologií [1]. Sketchpad představil revoluční možnost vytvářet v počítačovém prostředí kresby jinak, než za pomoci psaných příkazů. V rámci definovaného GUI bylo možné světelnou tužkou kreslit základní geometrické tvary, editovat dříve vytvořené výkresy, nebo instancionalizovat a následně kopírovat existující výkresy. Poprvé tak byl počítač představen jako nástroj s jehož pomocí lze uživatelsky přívětivou cestou tvořit výkresy technické dokumentace či umělecká díla.

Od té doby prošel CAD a CAM software rozsáhlou evolucí a pokořil několik významných milníků. Jedním z nich je zcela jistě přechod z kreslení dvojrozměrných geometrických tvarů na modelování trojrozměrných těles. To umožnilo na výkresech provádět složitější výpočty, generovat více či méně realistické vizualizace a výsledky lépe otestovat na přítomnost případných kolizí a nepřesností.

V dnešní době se CAD/CAM/CAE software využívá téměř ve všech odvětvích průmyslu, od projektování obytných i průmyslových staveb, přes práci s terénem, až po průmyslovou výrobu s využitím CNC strojů. Obuv, kterou dnes máte na nohou, auto, kterým jste se dostali do práce, nebo hrnek od kafe, který vám leží na stole, to vše jsou objekty, které byly navrženy a vyprodukovány s využitím CAD softwaru.

V posledních několika letech se výrobci CAD softwaru potýkají s novým milníkem, který je nutné pokořit. Tímto milníkem je přechod od kresby geometrických objektů k tvorbě „*informačního modelu*“. Tento nový přístup je v zásadě přechodem od tvorby oddělených výkresů a geometrických objektů k plně provázanému databázovému modelu, kdy každý prvek modelu (stavby) obsahuje vazby na řadu dalších konstrukčních a projektových objektů s nimiž se vzájemně ovlivňuje a doplňuje.

Tento princip byl za poslední dekádu a půl implementován řadou aplikací rozličných výrobců, z nichž každý přistupuje k informačnímu modelování v rámci svých specifik. Je nicméně zjevné, že i přes jednotlivé odlišnosti je informační modelování směrem, kterým se CAD průmysl ve stavebnictví ubírá a který bude i dále nabývat na významu.

Jedním z výrobců software pracujícího s informačním modelem budovy je americká korporace Autodesk, Inc., u které tuto roli vykonává rodina programů Revit. Revit nabízí několik rozdílných řešení určených pro jednotlivé fáze projektování staveb a pro odlišné části průmyslu - od návrhu nosných konstrukcí, přes modelování budovy, až po její vybavení vzduchotechnikou.

Revit Architecture je řešením umožňujícím architektům a projektantům vytvářet obecné návrhy budov a nad těmito provádět analýzy, vizualizace a podrobné dokumentace. Oproti předchozím CAD aplikacím společnosti Autodesk, zejména pak AutoCADu a AutoCADu Architecture, umožňuje Revit Architecture díky využití informačního modelování mnohem rychleji a přesněji provádět změny, které ovlivní celý projekt budovy a tímto způsobem se vyhnout nákladným kolizím a nepřesnostem v návrhu [2].

Součástí funkčnosti Revitu Architecture je tvorba legend vybraných prvků z modelu. Tyto legendy popisují parametry, skladbu a počty jednotlivých objektů a jsou nutnou součástí dokumentace většiny projektů. Součástí legend jsou schémata s různými pohledy a řezy daných objektů.

Požadavky na legendy jako součást dokumentace se však v rozdílných regionech světa velmi liší a zatímco pro americké uživatele jsou funkce nabízené Revitem dostatečné, striktnějším normám užívaným v evropském (a tedy i českém) stavebnictví již tak docela nevyhovují. Revit nabízí řešení formou manuálního kreslení vlastní legendy, kdy lze tímto způsobem zcela přizpůsobit její tvar i obsah. Tuto možnost nicméně nelze automatizovat a uživatele proto stojí extrémně mnoho času a úsilí legendy vytvářet a aktualizovat.

Součástí bakalářské práce je řešení tohoto problému formou nadstavby automatizující velkou část činností prováděných při tvorbě legendy. Tato nadstavba využívá aplikační programovací rozhraní Revitu, které je v práci stručně popsáno a zdokumentováno.

2 Informační model budovy

Než se pustíme do demonstrace naprogramované nadstavby a rozboru vlastního přínosu této práce, je vhodné na úvod prozkoumat technologii, na které jsou plugin a jeho mateřská aplikace Revit Architecture postaveny. Je to právě tato technologie, která nám vytváří mantinely a určuje, jakým způsobem nadstavba vnímá stavební prvky modelované budovy a pracuje s CAD výkresy. První kapitola je proto věnována popisu technologie, která je v českém prostředí známa pod jménem *Informační modelování budovy*.

Informační modelování budovy, neboli BIM [3], je dalším krokem ve vývoji CAD projektování staveb. Koncept technologie BIM však rozhodně není nový. V souvislostech, které přináší Informační Modelování Budovy, se uvažuje již více než třicet let a samotný termín „Building Information Modelling“ poprvé zazněl ústy Charlese Eastmana před bezmála dvaceti lety [4]. Je to však až nyní, kdy se tento koncept stává masovým a ve velkém měřítku přenáší do praxe.

Definovat ho lze z několika různých pohledů – jako soubor myšlenkových procesů architektů a projektantů, skupinu norem a požadavků na CAD software, nebo business model do něhož jsou zapojeny všechny strany zúčastněné na stavbě (investor, projekční kancelář, konstruktéři, subdodavatelé...) během celého jejího životního cyklu [5]. My v rámci této práce budeme na BIM pohlížet zejména jako na novou technologii implementovanou objemnou skupinou CAD programů od rozdílných výrobců.

2.1 BIM a CAD software

V CAD aplikacích starší generace uživatel běžně kreslí geometrické objekty reprezentující reálné prvky budovy. Tyto objekty mohou být jak dvojrozměrné (linie, polylinie, křivky...), tak trojrozměrné. Stále však zůstávají pouze geometrickými objekty a dokument, který s jejich pomocí uživatel vytváří, je jen sérií neprovázaných výkresů popisujících budovu. To, jaké konstrukční prvky jsou těmito geometrickými objekty reprezentovány, je určeno (uživatelem/programem) prostřednictvím jednoho, nebo více parametrů, které tyto objekty obsahují. Aplikace jiným způsobem nedokáže rozpoznat, zda je ten který konkrétní objekt zdí, potrubím nebo dveřmi.

Vzájemná neprovázanost vznikajících výkresů zároveň znamená, že pokud chce uživatel provést ve výkresu změnu, je nutné postupně editovat všechna místa, kde se měněný prvek vyskytuje. Příkladem může být posunutí vstupních dveří, nebo změna jejich rozměru. Výkresová dokumentace budovy však většinou obsahuje i řadu řezů, půdorysů a pohledů a uživatel proto musí vstupní dveře posunutou nebo zaměnit v každém z nich.

Informační modelování budovy se touto procesem téměř v ničem neliší. Uživatel již nekreslí linie a křivky, ale modeluje a popisuje budoucí stavbu. Svým způsobem se tato činnost podobá vyskládávání budovy ze stavebních kostek populární stavebnice Lego. Místo kreslení čar a bodů uživatel pokládá objekty, které jsou již na úrovni vnitřní databáze dané aplikace definovány jako zdi, dveře, pilíře, okna, kusy nábytku a další konstrukční prvky.

Každý objekt má své parametry dané materiálem, mezinárodními normami a programovou nebo uživatelskou databází prvků. Mezi parametry jednotlivých prvků patří

i jejich kompletní geometrická reprezentace. Při „nakreslení“ stěny se proto o většinu geometrických i parametrických detailů stará samotná aplikace. Šířka stěny je dána jejím složením, výška je dána normami a případně upravena dle navazujících prvků ve vyšší horizontální hladině. Celý proces projektování se tak obrací - uživatel již nekreslí anonymní geometrické objekty, které následně pojmenovává, ale pokládá konkrétní konstrukční prvky a vykreslení jejich geometrie ponechává v režii aplikace.

S tím vším souvisí další důležitý aspekt BIM aplikací – **parametrické vazby**. Jelikož při projektování nevznikají již jen neurčité geometrické objekty, ale konkrétní konstrukční objekty, jsou mnohem jasněji definovány vztahy, které mezi jednotlivými prvky mohou existovat. Na základě těchto vztahů a vazeb je aplikací při provádění změn v projektu automaticky upravována geometrie a další vlastnosti modelu. Podrobněji si to vysvětlíme na následujícím příkladě.

Představme si rodinný dům o dvou podlažích propojených schodištěm vyprojektovaný v libovolném programu pracujícím s informačním modelem budovy. Po předvedení vizualizací modelu investorovi obdržíme požadavek na změnu výšky jednotlivých podlaží. Pokud bychom v této fázi měli k dispozici pouze 3D CAD aplikaci, která nevyužívá parametrické vazby objektů a vertikálně posunuli druhé podlaží, tato změna se do žádých jiných objektů nepromítne. Dostaneme tak model, kde stěny z prvního podlaží končí ve „vzduchoprázdnu“ a nenavazují na druhé podlaží, okna ve druhém patře se dotýkají podlahy a schodiště spojující jednotlivá podlaží končí pod hladinou druhého podlaží. Všechny tyto kolize a nepřesnosti pak musíme jako uživatelé manuálně jednu po druhé vyřešit.

Naproti tomu v BIM aplikaci je každý objekt definován mnohem podrobněji ve vazbě na model vznikající okolo něj. Každý prvek má proto řadu parametrických vazeb ke všem, pro něj důležitým, objektům v okolí. Při posunu podlaží se automaticky změní výška zdí, schodiště, umístění oken ve stěnách a další prvky tak, aby výsledný model zůstal *integritní*.

Parametrické vazby se nevztahují jen na vnější vztahy objektů, ale i na objekty umístěné v hierarchii níže. Pokud tedy budeme chtít na úrovni prvního podlaží do jedné ze stěn umístit nové vstupní dveře, aplikace automaticky upraví geometrii zdi a nové dveře do zdi zasadí tak, aby nevznikly žádné konflikty. Vlastnosti (váha, materiál...) vložených dveří od této chvíle ovlivňují parametry hierarchicky nadřazeného prvku – zdi. Při jakémkoliv změně na úrovni dveří se automaticky změní i parametry nadřazené zdi.

Ze všech výše vyjmenovaných vlastností BIM vyplývá snad nejdůležitější prvek této technologie – *možnost analýzy modelu* na základě vnitřích dat objektů z nichž je model vytvořen. Ještě dřív, než je modelovaná budova postavena, můžeme analyzovat její chování v nasimulovaných situacích. Protože aplikace ví, z čeho je každá stěna a každý nosný sloup zkonstruován, je možné model podrobit zátěžovým testům, teplotním analýzám, energetickým simulacím aj. V době, kdy ekologické myšlenky a požadavky na tzv. „*green building*“ sílí na významu, je tato vlastnost žádaná a často využívaná.

Ze stejného principu vychází možnost automaticky nalézt všechny konflikty, které se v modelu nacházejí, nebo tvořit výpisy (legendy) použitých materiálů a prvků.

2.2 Definice informačního modelu budovy

Abychom mohli aplikaci označit jako *podporující BIM*, je nutné aby v ní tvořené objekty a modely splňovaly několik pravidel [5]:

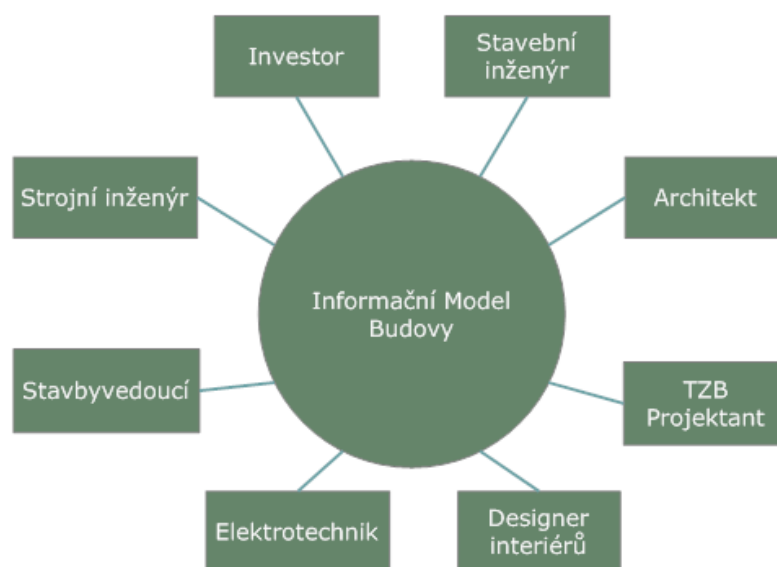
- Model se skládá z objektů definovaných geometricky a parametricky
- Geometrie je *neredundantní* a *konzistentní*. Z jednoho modelu budovy se generují všechny dvoj- i trojrozměrné pohledy, půdorysy, řezy, výkazy a další. Jakákoliv změna provedená v kterémkoliv pohledu se promítne do modelu.
- Parametrické vazby automaticky upravují geometrii odpovídajících objektů při vkládání nových prvků do modelu nebo změně již existujících objektů.
- Objekty mohou být definovány na kterémkoliv stupni hierarchie. Změna parametrů prvku na vyšším stupni ovlivní vlastnosti prvků mu podřízených. Z modelu se dají exportovat sady atributů a dat, např. použité stavební materiály, energetická data a jiné.

2.3 BIM aplikace

Programů, které nějakým způsobem pracují nad technologií BIM, je nespočet a svou komplexitou se řadí od jednoduchých nadstaveb pro ekologické analýzy a výpočty, až po obsáhlé balíčky programů určených pro trojrozměrné projektování budov, jejich vybavení TZB prvky, umístění modelu do reálného terénu a tvorbu fotorealistických vizualizací. Naším cílem v rámci této práce není je všechny vyjmenovat, ale uvést ty nejvýznamnější programy, které jsou považovány za průkopníky BIMu a dnes oblasti projektování informačního modelu dominují.

Přechod k technologii BIM byl historicky pro většinu výrobců pozvolným procesem a proto existuje skupina aplikací, která je mezikrokem od 3D projektování geometrických objektů k tvorbě provázaného informačního modelu. Typickým příkladem může být program **AutoCAD Architecture**, který sice pracuje nad modelem složeným z konstrukčních prvků, ale dosahuje toho na programovém jádře AutoCADu. To tento proces v mnohém limituje a neumožňuje udržovat provázanost prvků na takové úrovni jako v novějších aplikacích, které jsou již od prvotního návrhu cílené na proces informačního modelování (více o AutoCAD Architecture viz. kapitola 2.3.3). Zařazení těchto programů do, nebo mimo škatulku BIM software je z toho důvodu mnohdy sporné a i v rámci této práce proto bude prezentován pouze jeden z možných pohledů na tuto problematiku.

Relativní definovanost požadavků na BIM projektování má za důsledek vznik jednotného normovaného datového modelu, který podporuje většina aplikací napříč rozdílnými výrobci a zaměřenými. Tento model se nazývá **IFC (Industry Foundation Classes)** a v rámci své specifikace definuje tři rozdílné souborové formáty pro sdílení modelových dat mezi odlišnými aplikacemi [6]. Tyto formáty fungují jako hlavní pojítka mezi všemi BIM programy.



Obrázek 1: Propojení profesí v rámci informačního modelu budovy

2.3.1 Graphisoft ArchiCAD

Maďarská společnost Graphisoft se zabývá vývojem CAD softwaru pro architekturu a stavebnictví již od svého vzniku počátkem osmdesátých let. V roce 1984 vydal Graphisoft první verzi **ArchiCADu** (v té době pod jménem *Radar CH*) - programu, který se v roce 2010 dočkal své čtrnácté verze [7]. Byl to právě **ArchiCAD**, který již od prvních verzí v osmdesátých letech představoval zárodky principů BIM v rámci svého konceptu *virtuální budovy*. Základem byla databáze virtuálních stavebních prvků, definovaných svou geometrií a sadou konstrukčních dat a parametrů, ze kterých mohl uživatel vybírat a tvořit svůj model. Tím ArchiCAD výrazně předběhnul svou dobu a odlišil se tak od většiny konkurenčních 2D a 3D CAD nástrojů, které do té doby pracovaly stále jen s čistě geometrickými objekty a sadami neprovázaných výkresů.

K **ArchiCADu** existuje řada Graphisoftem vytvořených rozšíření, které s virtuálním modelem budovy pracují z úhlů pohledu odlišných profesí podílejících se na stavbě, viz. Tabulka 1 [7]. Nejinnovativnějším z nich je **Graphisoft BIM Server**, který slouží jako centrální repozitář informačního modelu budovy nad nímž pracuje velké množství CAD aplikací z různých míst v síti.

2.3.2 Bentley

Hlavním softwarovým pilířem společnosti Bentley Systems, Incorporated je aplikace **Microstation**, jež byla poprvé uvedena na trh v osmdesátých letech a k dnešnímu dni existuje už ve své osmé verzi. **Microstation**, podobně jako například AutoCAD od konkurenční společnosti Autodesk, se stal základní platformou pro nespočet rozšiřujících programů specializovaných na konkrétní oblast stavebnictví.

Program	Použití
ArchiCAD	2D kreslení a 3D modelování nad jednotným informačním modelem pro architekty, designéry a projektanty. Tvorba dokumentace, vizualizací a analýz.
Graphisoft MEP Modeler	Rozšíření ArchiCADu umožňující doplnit model budovy o prvky vzduchotechniky a potrubní systémy.
Graphisoft BIM Server	Řešení umožňující hostování a sdílení virtuálních modelů budovy mezi různými místy v síti v rámci velkých projekčních týmů/kanceláří.
Graphisoft EcoDesigner	Rozšíření ArchiCADu pro provádění energetických analýz virtuálního modelu budovy.
Virtual Building Explorer	Nadstavba ArchiCADu pro tvorbu trojrozměrných interaktivních prezentací virtuálního modelu budovy.

Tabulka 1: BIM software Graphisoftu

Bentley nicméně nabízí i řešení na bázi informačního modelování budov. To je v tomto případě představeno jako „*integrováný projektový model*“, který je vyskládán z aplikačních modulů jednotlivých Bentley BIM programů (Bentley Architecture, Bentley Building Electrical Systems, Bentley Structure...) [8].

2.3.3 Autodesk Revit

Společnost Autodesk je známá především díky svému vlajkovému programu **AutoCAD** (a jeho jednodušší verzi **AutoCAD LT**), který inženýrům ze všech oblastí průmyslu slouží jako návrhový software již od roku 1982 [2]. V devadesátých letech začal Autodesk vytvářet specializované aplikace běžící na programovém jádru AutoCADu, zaměřené na konkrétní průmyslová odvětví. V rámci tohoto procesu vznikla řada programů pro oblasti strojírenství, stavebnictví a stavebního inženýrství.

Významným milníkem v oblasti stavebnictví je aplikace **Architectural Desktop** (dále jen ADT), kterou Autodesk představil v roce 1998 [2]. Dnes je tato aplikace známá pod jménem **AutoCAD Architecture** a v roce 2010 se dočkala již své třinácté verze. ADT obohatilo geometrické prvky AutoCADu o dodatečné konstrukční parametry a vztahy k některým okolním objektům. Příkladem budiž situace ve které uživatel vymaže objekt zdi, který v sobě má zasazených několik oken. Díky parametrickým vazbám jsou následně vymazána i tyto. Tato provázanost byla ale limitována tím, co umožňuje jádro AutoCADu, jež bylo navrženo a naprogramováno pro zcela jiný přístup ke kreslení a vzájemná asociativita objektů proto ani zdaleka není tak samozřejmá a obsáhlá jako u pravého BIM řešení.

Pravé BIM řešení Autodesk představil až v roce 2002, po odkoupení společnosti Revit Technology Corporation [2]. **Autodesk Revit Series**, běžící na své vlastní technologii nezávislé na AutoCADu, představil pracovní prostředí založené na práci s centrálním virtuálním modelem budovy z nějž jsou generovány všechny pohledy a výkazy a který

Program	Použití
Revit Architecture	Tvorba informačního modelu budovy, projektové dokumentace, vizualizací.
Revit Structure	Tvorba modelu doplněná o prvky pozemního stavitelství. Provádění analýz konstrukčních prvků, tvorba dokumentace.
Revit MEP	Tvorba modelu z pohledu vzduchotechniky, potrubních a elektrických systémů.
Civil 3D	Tvorba modelu terénu a projektování inženýrských staveb a infrastruktury.
Navisworks	Koordinace projektu, analýzy a řešení kolizí a nepřesností.
Ecotect Analysis	Analýza udržitelného návrhu. Vyhodnocení energetické náročnosti budovy.
Quantity Takeoff	Výpočet nákladů komplexních projektů.

Tabulka 2: BIM software Autodesku

nabízí jejich *obousměrnou provázanost*. Více o struktuře dat a aplikaci obecně si povíme v Kapitole 3.

Na bázi Revit Series je založena celá řada aplikací určených pro rozdílná odvětví stavebního průmyslu (viz. Tabulka 2).

2.3.4 Ostatní

Samozřejmě existuje celá řada dalších méně významných aplikací a nadstaveb pro dílčí práci nad informačním modelem budov. Pokud se tyto aplikace jeví jako perspektivní, jsou většinou rychle skoupeny jednou z výše uvedených společností a zařazeny do jejího portfolia (např. Revit Technologies, Navisworks JetStream a další...).

Za zmínku také rozhodně stojí **Google SketchUp**, jež je šířen zdarma, uživatelům umožňuje tvorbu 3D modelů a jejich umístění do prostředí Google Earth a jehož modely jsou v dnešní době již do určité míry podporovány ve všech hlavních BIM aplikacích.

Mezi další významné aplikace patří **Tekla Structures** pro projektování trojrozměrných konstrukčních modelů, programový balíček **GRAITEC Advance** pro modelování ocelových a betonových konstrukcí, nebo **Nemetschek VectorWorks Architect**.

3 Autodesk Revit Architecture

Jak jsme si již řekli v předchozí kapitole, Autodesk Revit Architecture je CAD aplikací pro projektování informačního modelu budovy se vším, co k tomu patří. Model budovy je základním stavebním kamenem kolem kterého se točí veškerá funkcionální programy. Z modelu vyplývají vztahy mezi jednotlivými prvky projektu, jsou z něj generovány pohledy a výpisy a je do něj promítnuta každá změna provedena na libovolném místě v projektu.

3.1 Organizace dat

Informační model budovy je uložen ve formě singulárního databázového souboru s příponou .RVT. Každý model je složený z libovolného počtu stavebních prvků - zdí, oken, dveří, podlaží aj. - které jsou definovány svou *rodinou* a *typem*. Definice *rodin* jsou uloženy v souborech s příponou .RFT (Autodesk Revit Family Type File) z nichž jsou načítány do modelu.

Každý objekt použitý v projektu náleží do hierarchie, která je popsána termíny *element* nebo *kategorie*, *rodina*, *typ* a *instance* [9]. Toto rozdělení se netýká jen fyzických stavebních prvků, ale i objektů jakými jsou pohledy nebo výpisy. To je důležité, protože uživatel může editovat parametry objektů na každé z popsaných úrovní.

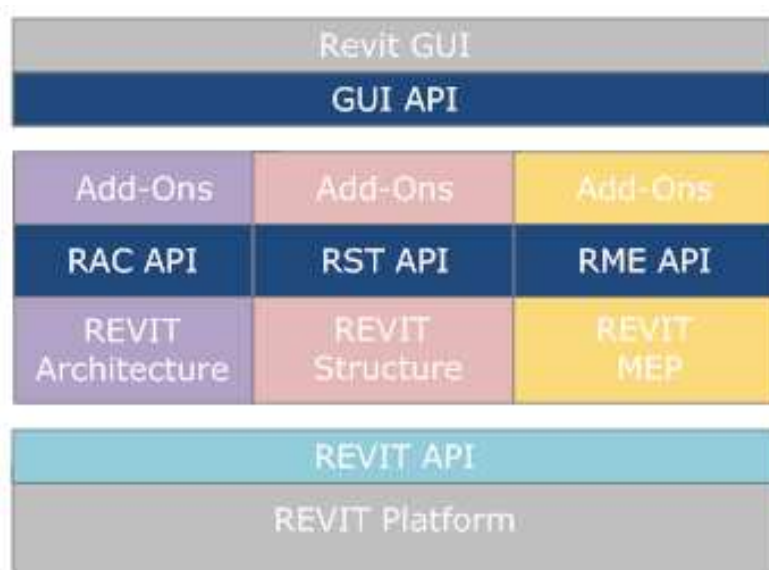
- **Element/kategorie:** Všechny objekty v modelu jsou přiřazeny široce definované kategorii. Příkladem kategorie jsou okna nebo pohledy.
- **Rodina:** Rodina sdružuje prvky s podobnou geometrií. Prvky kategorie *okno* mohou být dále rozděleny do rodin oken kruhových, lomených, trojúhelníkových apod.
- **Typ:** Typ definuje parametrické vlastnosti objektu, způsob jeho grafické reprezentace v rozdílných pohledech a to jak ovlivňuje a je ovlivňován okolními objekty. Okna z rodiny obdélníkových oken mohou být rozdělena na typy podle rozměrů, např. 406 x 610mm, 1180 x 1350mm a další.
- **Instance:** Instance je jeden konkrétní objekt určitého typu umístěný do modelu.

3.2 Podpora nadstavb

Stejně jako všechny ostatní aplikace Autodesku (nejznámější je v tomto ohledu AutoCAD) nabízí i Revit Architecture vývojářům možnost přistupovat k vnitřní databázi modelu a využívat ji při programování nadstavb. S využitím Revit API je možné z modelu získávat geometrická i parametrická data, tvořit a editovat stavební prvky, automatizovat dílčí úlohy v rámci obsáhlých maker, využívat data obsažená v databázi modelu vnějšími aplikacemi mimo prostředí Revitu, provádět analýzy a tvořit dokumentaci k existujícím projektům.

3.3 Revit Architecture API

Revit Platform API je souborem veřejně přístupných metod umožňujících interakci s vnitřní databází Revitu. Protože na programovém jádru Revitu dnes existují tři rozdílné aplikace (viz. Obrázek 2) [10], má i Revit API tři v detailech se lišící varianty. Tyto varianty využívají shodně definovaný informační model i pracovní prostředí aplikace, ale obsahují metody pro práci s odlišnými rodinami prvků, jejich vlastnostmi a analýzami. My budeme v rámci této práce nadále mluvit jen o API použitém v Revitu Architecture 2011 (verze je důležitá z důvodu změny názvů některých tříd a jmenných prostorů mezi verzemi 2010 a 2011).



Obrázek 2: Struktura Revit API

Revit API se skládá ze dvou dynamických knihoven - **RevitAPI.dll** a **RevitAPIUI.dll** - jejichž metody pracují pouze nad právě spuštěnou instancí programu (jinými slovy *nelze ovlivňovat vnitřní databázi Revitu jinak, než z prostředí právě běžící instance Revitu*). Tyto knihovny jsou uloženy v kořenovém adresáři programu.

RevitAPI.dll obsahuje metody pro přístupu k dokumentům (projektům), nastavení právě běžící instance aplikace, elementům a všem jejich parametrům.

RevitAPIUI.dll obsahuje metody definující uživatelské rozhraní Revitu, včetně definice pásu karet (tak je v české verzi pojmenován panel nástrojů, jehož vizuální styl kopíruje uživatelské rozhraní Microsoft Office verzí 2007 a 2010), dotazovacích oken aplikace a rozhraní *IEternalCommand* a *IEternalApplication* (k těm se vrátíme v Kapitole 3.5.2).

K metodám obsaženým v Revit API je možné přistupovat prostřednictvím programovacích jazyků platformy Microsoft .NET Framework (verze 3.5). Většina uživatelské dokumentace Revit API je psána z pohledu programovacích jazyků C# a Visual Basic

.NET. To je velkou změnou oproti AutoCADu, který pro programování a integraci nadstavby využívá zejména jazyku Visual LISP.

Všechny nadstavby použité v Revitu musí být zkompileovány do podoby dynamických knihoven (souborů s příponou .dll) a musí být pouze **jednovláknové**. V současné době Revit nepodporuje ovlivňování běhu programu vícevláknovými nadstavbami.

3.4 Umístění nadstaveb do prostředí Revitu

Naprogramované nadstavby jsou od verze 2011 do samotného programu integrovány na základě tzv. „*manifest file*“ - XML dokumentů s příponou .addin, které popisují a odkazují na všechny knihovny, které se mají během spouštění Revitu zpracovat a zařadit do aplikace. Umístění *manifest files* je v závislosti na operačním systému následující:

- **Windows XP:** C:\Documents and Settings\[user]\Application Data\Autodesk\Revit\Addins\2011\
- **Vista/Windows 7:** C:\Users\[user]\AppData\Roaming\Autodesk\Revit\Addins\2011\

3.5 Hello World s využitím Revit API

Průběh programování nadstavby pro Revit, její následné zařazení do prostředí aplikace a všechna potřebná nastavení si podrobně vysvětlíme na příkladu pluginu, který v aktivním uživatelském prostředí Revitu vytvoří dialogové okno s textem "Hello World". Současně si blíže vysvětlíme některé základní prvky, bez kterých se žádná nadstavba Revitu neobejde a které jsou nezbytnou součástí implementační části této práce (použitá rozhraní, XML manifesty, transakční atributy...), ale ke kterým se již v kapitole věnované implementaci nebudeme vracet.

Plugin budeme tvořit v programovacím jazyce C# v prostředí Microsoft Visual C# 2008 Express Edition.

3.5.1 Tvorba nového projektu

Prvním krokem k námi požadované nadstavbě Hello World je vytvoření nového projektu v prostředí Visual Studio.

- Pro nový projekt zvolíme šablonu **C# Class Library** a pojmenujeme ho HelloWorldProject.
- Nyní potřebujeme našemu projektu přiřadit reference na Revit API a metody v něm obsažené, které budeme v naší nadstavbě volat. V Solution Exploreru najdeme složku References a příkazem „**Add Reference...**“ přidáme novou.
- V nově otevřeném dialogovém okně se přepneme na záložku Browse a v adresářové struktuře najdeme a vybereme RevitAPI.dll a RevitAPIUI.dll (umístěné v kořenovém adresáři instalace Revitu).

- Abychom se ujistili, že naše nadstavba nebude vytvářet duplikáty Revitovských tříd, změníme u obou referencí (v rámci jejich properties) vlastnost **Copy Local** z true na false.

3.5.2 Třída HelloWorld

V nově vzniklém projektu je od samého počátku vytvořena třída Class1.cs, které se budeme nyní věnovat. Toto je jediná třída, kterou v našem projektu budeme potřebovat a ze které budeme volat metody rozhraní Revitu. Pro usnadnění orientace a práce na projektu ji přejmenujeme na HelloWorld.cs. Na úvod je vhodné importovat jmenné prostory, jejichž metody budeme při programování nadstavby využívat (viz. Výpis 1).

```
using Autodesk.Revit.DB;
using Autodesk.Revit.UI;
```

Výpis 1: Jmenné prostory

Pokud má být třída HelloWorld akceptována Revitem, je nutné aby byla definována jako „externí aplikace“ nebo „externí příkaz“ a implementovala příslušné rozhraní.

Externí příkaz (implementující rozhraní IExternalCommand) je určen pro jednodušší nadstavby, které provádí menší množství operací nad uživatelem vybranými nebo nadstavbou definovanými elementy. Objekt externího příkazu je Revitem držen v paměti jen po dobu vykonávání příkazu. Po jeho ukončení se Revit objektu a všech jeho vnitřních dat zbavuje. V uživatelském prostředí Revitu je zařazen do pásu karet *Add-ins* jako součást nabídky vysouvacího panelu *External tools*.

Externí aplikace (implementující rozhraní IExternalApplication) je naopak určena pro složitější nadstavby. Obsahuje metody OnStartup() a OnShutdown() volané při spuštění a ukončování samotného Revitu. Narozdíl od externího příkazu je objekt externí aplikace Revitem držen v paměti dokud není daná instance Revitu ukončena. Externí aplikace jsou graficky zobrazeny jako nové pásy karet a jejich součástí může být jeden, nebo více externích příkazů

Pro naši nadstavbu nám bohatě postačuje definice externího příkazu. HelloWorld bude proto implementovat rozhraní IExternalCommand. Toto rozhraní definuje metodu Execute(), která je volána při každém spuštění nadstavby a vrací hodnoty typu *Autodesk.Revit.UI.Result* **Succeeded**, **Failed** a **Cancelled**.

Po implementování IExternalCommand a metody Execute() nám nezbývá nic jiného, než vytvořit dialogové okno prostřednictvím třídy Autodesk.Revit.UI.TaskDialog a metody Show(„Název titulku“, „Zpráva dialogového okna“) a aplikaci vrátit hodnotu Succeeded značící úspěšné provedení a dokončení externího příkazu.

Posledním krokem je doplnění takto napsaného příkazu o dva povinné atributy, které určují jakým způsobem bude nadstavba interagovat s databází a dokumenty Revitu. Těmito atributy jsou **transakce** a **regenerace**. **Transakce** ovlivňuje komunikaci s vnitřní databází revitu a určuje, kdy se v rámci běhu nadstavby provádí na databázi příkaz *commit*. Nejběžnějším přístupem je nastavit transakci na *automatickou* a tím nechat Revit API potvrdit všechny změny na databázi vždy až na konci běhu pluginu.

Regenerace je nastavením toho, jak často je aktivní dokument aktualizován a překreslován. Protože jsou projekty Revitu často velmi rozsáhlé, každé překreslení výkresů znamená nezanedbatelné čekání na straně uživatele. Je proto doporučeno v rámci programování nadstaveb využívat zásadně *manuální* hodnotu tohoto atributu a regenerace provádět konkrétním voláním na vhodném místě vlastní nadstavby.

```
namespace HelloWorldProject
{
    [Autodesk.Revit.Attributes.Transaction(Autodesk.Revit.Attributes.TransactionModeAutomatic)]
    [Autodesk.Revit.Attributes.Regeneration(Autodesk.Revit.Attributes.RegenerationOption.Manual)]
    public class HelloWorld : IExternalCommand
    {
        public Result Execute(ExternalCommandData commandData,
            ref string message, ElementSet elements)
        {
            TaskDialog.Show("Title", "Hello World!");
            return Autodesk.Revit.UI.Result.Succeeded;
        }
    }
}
```

Výpis 2: Třída HelloWorld

3.5.3 Začlenění do Revitu

Dalším krokem je zkompileování projektu. V nastavení projektu, v záložce *Build* můžeme nastavit cílový adresář do kterého budou zkompileované soubory umístěny. Pro jednoduchost změníme cílový adresář na „C:\RevitApiDlls\“, v záložce *Application* zkontrolujeme, zda používáme odpovídající verzi .NET Frameworku (3.5) a náš projekt zkompilejeme (klávesa F6).

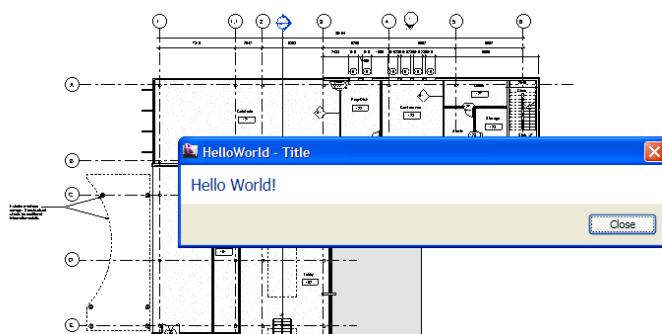
```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<RevitAddIns>
  <AddIn Type="Command">
    <Assembly>C:\RevitApiDlls\HelloWorldProject.dll</Assembly>
    <AddInId>f4bf069f-a444-405e-b3de-42b81bb2bf61</AddInId>
    <FullClassName>HelloWorldProject.HelloWorld</FullClassName>
    <Text>HelloWorld</Text>
  </AddIn>
</RevitAddIns>
```

Výpis 3: HelloWorld manifest

Nyní, když už máme v rukou hotový, zkompileovaný program, je nutné říct Revitu z jaké lokace ho má při každém startu aplikace načítat. V libovolném textovém editoru vytvoříme soubor s příponou **.addin** a XML strukturou popsanou ve Výpisu 3 a tento soubor umístíme do lokace „C:\Documents and Settings\All Users\Application Data\Autodesk\Revit\Addins\2011\“.

XML Tag	Význam
AddIn	Ohraničuje každý unikátní externí příkaz (Type="Command") a externí aplikaci (Type="Application") odkazované v .addin souboru.
Assembly	Umístění zkompilované knihovny pluginu.
AddInId	Unikátní GUID identifikace pluginu. Je nutné, aby každý externí příkaz a každá externí aplikace běžící současně v rámci jedné instance Revitu měla unikátní GUID označení. Možnost vygenerovat na řadě webových stránek on-line.
FullClassName	Celé jméno třídy obsažené ve zkompilované knihovně pluginu, která implementuje rozhraní IExternalCommand nebo IExternalApplication.
Text	Pojmenování tlačítka spouštějícího plugin v prostředí Revitu.

Tabulka 3: XML struktura souboru HelloWorld.addin



Obrázek 3: Hello World v prostředí Revitu

4 Revit Legendy+

Součástí této práce je plugin **Legendy+** umožňující tvorbu dynamických legend v prostředí Revitu Architecture 2011. Ačkoliv Revit umožňuje tvorbu výkazů a legend, trpí v této oblasti faktem, že je prodáván a používán na celém světě. Z toho důvodu jsou při jeho vývoji dělány mnohé kompromisy, jedním z nichž je právě zjednodušené prostředí pro tvorbu legend. A ačkoliv je toto prostředí naprosto vyhovující pro například americký trh (kde jsou požadavky na legendy jiné, než v Evropě), v rámci českých norem a zvyklostí je nedostatečné.

Funkce, které Revit nabízí pro export materiálů a prvků modelu a tvorbu legend, jsou na výstupu prezentovány holými tabulkami u kterých není možné měnit geometrii, či přiřazovat schémata jednotlivých prvků. Takové legendy jsou poté pro využití v českém stavebnictví nepoužitelné.

Výkaz oken							
Označení typu	Popis	Šířka	Výška	Výška par	Podlaží	Počet	Cena
18	popis okna 1	915	1830	915	Podlaží 1	1	1001.00
45	popis okna 2	915	1525	915	Podlaží 1	1	1002.00
40	popis okna 3	915	1830	915	Podlaží 1	1	
19	popis okna 4	1220	1220	915	Podlaží 1	1	
15	popis okna 5	915	1220	915	Podlaží 1	1	
44	popis okna 6	925	1220	915	Podlaží 1	1	1006.00

Obrázek 4: Výkaz generovaný základními funkcemi Revitu

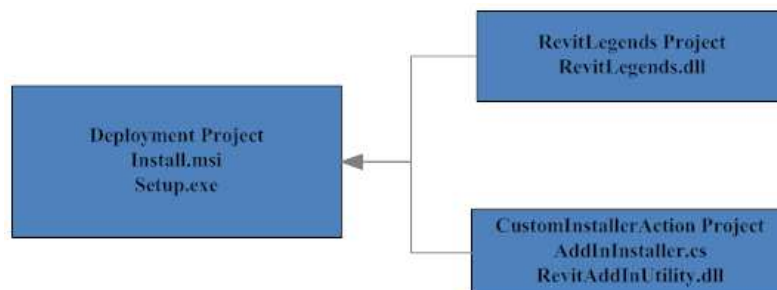
V průběhu mnohaletého vývoje byl tento nedostatek Revitu částečně napraven v rámci tzv. „*pohledu*“ pro Legendy. Tento pohled je oddělenou uživatelskou plochou, která není přímo svázána s modelem a která je určena pro kreslení geometricky libovolných tabulek se schémata a texty. Zároveň má však zásadní nedostatek a tím je chybějící automatizace kreslení. Každou buňku legendy a každé schéma musí uživatel na plochu nanést manuálně. Stejně tak je potřeba přepsat každý vložený text tak, aby reflektoval objektový parametr z modelu. Takový postup je pracný a u jakýchkoliv větších projektů nevhodný.

Z možností nabízených Revitem v *pohledu pro legendy* vychází plugin Legendy+. Jeho cílem je umožnit uživateli vytvořit návrh budoucí legendy a dle tohoto návrhu automaticky vygenerovat kompletní tabulku, včetně načtení všech odpovídajících parametrů z modelu a zobrazení požadovaných schémat. Na základě existujícího návrhu je možné vytvořit libovolný počet legend, které lze obdobným způsobem velmi rychle aktualizovat a doplnit o nové údaje. Tímto způsobem činí plugin z několikahodinové operace otázku několika vteřin a pár kliknutí myši.

4.1 Struktura pluginu

Celý projekt pluginu se před svou kompilací skládá ze tří komponent: knihovny pluginu; deployment projektu, který se stará o vytváření instalačního balíčku; a instalačního

skriptu `AddInInstaller.cs`, který obstarává generování XML manifest souborů a jejich odebrání při odinstalaci pluginu.



Obrázek 5: Projektová struktura Legendy+

4.1.1 Instalace a distribuce

Legendy+ je distribuován formou balíčku instalační služby systému Windows (soubor s příponou `.msi`) a spustitelného `.exe` souboru, který balíček spouští. Tento balíček nabízí uživateli obvyklé možnosti nastavení instalace a kromě samotného pluginu se stará i o jeho registraci do Revitu prostřednictvím tvorby XML manifestu ve složce určené pro data aplikací. K tomuto účelu využívá knihovny **AddInManager.dll**, která je obsažena ve volně šířeném Revit SDK a obsahuje metody pro zjištění všech instalací Revitu na daném počítači, určení cest k adresářům s daty aplikací a automatické generování XML manifestů.

Při použití pluginu v prostředí firemních sítí je nutné povolit uživatelům využívajícím plugin vytvářet a editovat soubory umístěné v kořenovém adresáři nadstavby (běžně umístěn ve složce `Program Files/Adeon/LegendyPlus/`). V tomto adresáři se za běhu programu vytváří a upravuje soubor `"RevitLegendsParam.txt"`, který je nezbytný pro správné fungování nadstavby (více o tomto souboru v kapitole 4.3.1).

4.1.2 Programové jádro

Jak jsme si již vysvětlili v kapitole 3.5.2, každá nadstavba Revitu musí implementovat jedno ze dvou nabízených rozhraní. V případě Legendy+ jsou využity rozhraní obě a to tak, aby bylo možné používat více samostatných příkazů v rámci jedné nadstavby.

Programové jádro nadstavby Legendy+ se skládá z následujících součástí:

- Třídy `RevitLegend.cs` implementující rozhraní `IExternalApplication`. Tato třída spravuje a registruje jednotlivé příkazy provádějící izolované funkce. Zároveň obstarává tvorbu a ovládání grafického panelu pro plugin v rámci tzv. "Páso karet" Revitu.
- Tříd `AddAttribute.cs`, `DisconnectParameters.cs`, `ShowAttributes.cs` a `CreateLegendCommand.cs` implementujících rozhraní `IExternalCommand`. Každá z těchto tříd

reprezentuje jeden příkaz spustitelný v prostředí Revitu. V uživatelském rozhraní jsou tyto příkazy reprezentovány tlačítkem v nabídce funkcí pluginu.

- Tříd obsažených ve složce Forms, které jsou grafickými okny (System.Forms) pro komunikaci s uživatelem v různých fázích běhu nadstavby. Tyto formuláře komunikují s jimi nadřazenými externími příkazy a provádějí triviální filtrovací operace nad databází Revitu.
- Pomocných tříd složky Family Type Holding Classes. Ty jsou velmi jednoduchými objekty, které seskupují vlastnosti rodin a jejich instancí a umožňují tyto seskupení zobrazovat a dále s nimi pracovat v rozdílných formulářích.
- Tzv. "resources" souborů, které slouží pro lokalizaci pluginu v závislosti na jazykové verzi nainstalované verze Revitu. Jedná se pouze o lokalizaci ovládacích prvků nadstavby, veškeré vnitřní volání programu je nezávislé na aktuální jazykové verzi. Nadstavba obsahuje lokalizace pro český a anglický jazyk (ten je využíván při detekci jakékoliv nečeské verze Revitu).

4.2 Algoritmus tvorby legendy

Tvorba legendy v rámci nadstavby Legendy+ automatizuje velké množství operací, které jsou běžně v prostředí Revitu uživateli dostupné v rámci speciálního tomu vyhrazenému pohledu. Původní idea za tvorbou nadstavby byla úplná automatizace tohoto procesu na základě uživatelem zadaných nastavení (počet sloupců a řádků, parametry, které mají být v tabulce obsaženy, atd.). Nicméně takový přístup se ukázal být příliš svazujícím a nevyhovujícím zvyklostech uživatelů softwaru společnosti Autodesk (mezi které patří i Revit). Finální verze Legendy+ proto ponechává část té nejzákladnější práce na uživateli.

Základem je **vzor** (v řeči Revitu "předpis") tabulky, který uživatel připraví před samotným spuštěním nadstavby. Tento vzor je jednou buňkou (sloupcem, nebo řádkem) budoucí tabulky a zobrazuje jak bude legenda vypadat z hlediska své geometrie a rozměrů, parametrů, které bude zachycovat a schémat, které bude obsahovat. Tento vzor se skládá z třech typů Revitovských objektů: *detailevých čar*, *textových poznámek* (pro nadpisy sloupců tabulky a zachycení samotných parametrů) a *komponent legendy* (schémat).

Po vytvoření vzoru legendy uživatelem je zapotřebí označit, které textové poznámky slouží pro reflektaci parametrů modelu a které jsou pouhými nadpisy a pomocnými texty, jejichž obsah se nebude během procesu generování tabulky měnit. Pro toto se využívá příkaz nadstavby **Připojit atribut** (více o něm v kapitole 4.3.1). Po nastavení propojení mezi textovými poznámkami a parametry uživatel vybere, pro které objekty (typy rodin) chce legendu nechat vytvořit a následně dochází k samotnému vygenerování tabulky. Tento proces se skládá ze dvou stejně důležitých součástí: získání a protřídění kolekce modelových objektů, které budou v legendě zobrazeny a vykreslení a naplnění samotné tabulky.

Získání modelových objektů z aktivního projektu Revitu se provádí použitím filtrů, které jsou součástí rozhraní Revit API. Nabízené filtry jsou dvojího typu: "*pomalé*" a "*rychlé*". *Rychlé* filtry se vyznačují tím, že operují nad velmi omezeným objektovým

rozhranním, které umožňuje načítat pouze ty nejzákladnější atributy filtrovaných elementů. Ostatní dostupné vlastnosti jsou načítány jen u těch elementů, které úspěšně projdou filtrem. Díky tomu jsou tyto filtry pamětově i časově velmi úsporné. Mezi tyto filtry patří například třídy *ElementCategoryFilter* (pro filtrování určitých kategorií/typů rodin) a *ElementIsElementTypeFilter* (pro přefiltrování kolekce na instancní prvky a typy rodin/kategorie), které jsou základem práce nadstavby.

Pomalé filtry naopak načítají každý filtrovaný element včetně všech jeho parametrů do paměti a až nad těmito objekty provádějí samotnou filtraci. Patří mezi ně například *ElementParameterFilter* (pro filtraci elementů dle zvoleného parametru), nebo *FamilyInstanceFilter* (pro filtraci instancí konkrétní rodiny). Tyto filtry jsou náročné na výpočetní čas i potřebnou velikost paměti a v nadstavbě Legendy+ se proto nevyskytují.

Použitím rychlých filtrů získává nadstavba několik kolekcí objektů z aktivního projektu Revitu: kolekci všech **typů rodin** (například unikátních typů zdí), kolekci všech **instancí** dané rodiny, kolekci **schémat** a kolekci **textových poznámek** použitých v legendě. Tyto kolekce jsou předávány jako objekty typu *System.Collections.List*.

Následuje řada operací prováděných nad těmito množinami za pomoci integrovaného dotazovacího jazyka LINQ. Jsou propojeny kolekce typů rodin a instancí a na základě tohoto nově vzniklého listu jsou vyřazeny všechny typy rodin, které se sice nacházejí v databázi rodin projektu, ale nejsou v samotném modelu zastoupeny žádnými instancemi. Výsledkem je kolekce/tabulka všech objektů (včetně parametru počtu jejich instancí), které budou v legendě zastoupeny.

Nyní je nadstavbě známo kolik prvků bude nově vznikající legenda monitorovat a tedy kolik řádků (sloupců) bude budoucí tabulka obsahovat. Může proto započít proces tvorby této tabulky. Základem je *výběr* (*Autodesk.Revit.UI.Document.Selection*), který měl uživatel aktivní před spuštěním nadstavby. Tento výběr obsahuje celý předpis tabulky, nebo jeho část, která má být pro tvorbu tabulky použita.

Bohužel z důvodů omezení Revit API není možné tento výběr začít okamžitě rozkopírovávat a vytvořit tak kompletní tabulku. Některé objekty (zejména schémata) nenabízejí v aktuální verzi API (v.2011) funkčnost pro vytváření nových instancí, nebo jejich kopírování. Legendy+ tento problém řeší dočasným vytvořením takzvané *skupiny*, která v Revitu označuje libovolné uskupení objektů a rodin, které se v rámci projektu chovají jako jednolitý objekt. Se skupinami již lze v rámci Revit API provádět kopírovací operace a je proto možné předpis tabulky rozmnožit a po zrušení dočasných skupin získat zpět všechny individuální objekty, které tabulka obsahuje (viz. Výpis 4).

Při samotném kopírování předpisu (respektive skupiny ho obsahující) je nutné mít na vědomí vztahy mezi vnitřními a projektovými jednotkami Revitu. Všechny vnitřní databáze Revitu totiž vždy a za každých okolností využívají imperiální systém jednotek a tedy v případě rozměrů tabulky pracují s délkovými hodnotami v palcích a stopách. Jelikož je ovšem nadstavba určena zejména pro evropské uživatele (z důvodu striktnějších norem), většina vstupních i výstupních údajů legendy musí nutně být jednotky soustavy SI (metrické). Je proto nezbytně nutné vždy před každým zadáním jednotek pro práci nadstavby provést převod mezi projektovými jednotkami a vnitřními jednotkami Revitu.

Pro tento účel obsahuje Legendy+ několik převodních konstant (více viz. dokumentace nadstavby).

```
private bool createLegendTable(UIDocument uiDoc, Document doc, string direction, double
    distance, int numberOfCells)
{
    SelElementSet sel = uiDoc.Selection.Elements;
    Group selGroup = doc.Create.NewGroup(sel);
    Options options = doc.Application.Create.NewGeometryOptions();
    LocationPoint origin = selGroup.Location as LocationPoint;
    XYZ p = origin.Point;
    for (int i = 1; i < numberOfCells; i++)
    {
        XYZ copiedGroup = new XYZ(p.X, p.Y - ((height + distance / FEET_TO_MM) * i), p.Z);
        //Promenna height urcuje vysku jednoho radku legendy. V tomto zjednoduseni neni
        //jeho vypocet rozvinuty
        Group newGroup = doc.Create.PlaceGroup(copiedGroup, selGroup.GroupType);
        ElementSet eSet = newGroup.Ungroup();
        selGroup.Ungroup();
    }
    return true;
}
```

Výpis 4: Zjednodušená podoba algoritmu tvorby tabulky rozkopírováním vzoru

Po vytvoření tabulky rozkopírováním dochází ke spočítání počtu textových poznámek s přiřazenými parametry. Jejich vydělením celkovým počtem monitorovaných typů rodin získáváme počet iterací, kterými bude muset nadstavba projít pro vypsání všech požadovaných parametrů každého typu rodiny. Dochází k postupnému procházení a přepisování textových poznámek na parametry, jež mají reflektovat. V rámci tohoto procesu je dohlíženo na správnou interpretaci parametrů, které mohou být uloženy pěti rozdílnými datovými typy (Integer, ElementID, String, Double, None), z nichž ne všechny jsou tvaru, který by byl čitelný pro uživatele. Například hodnoty Integer jsou pouze označení enumerací, jejichž název je tím, co udává hodnotu parametru tak, jak ji vidí uživatel.

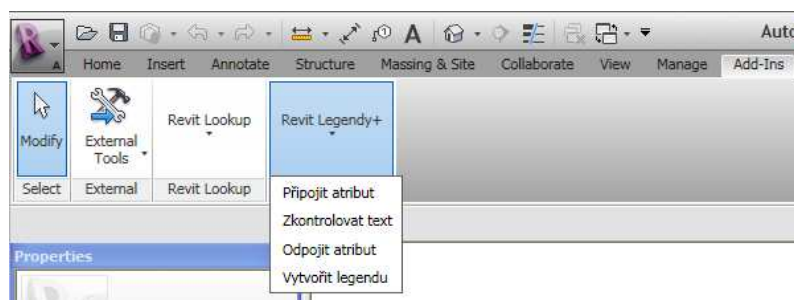
Následně dochází ke stejnému procesu přiřazování i u schémat a vzniká finální podoba legendy.

4.3 Legendy+ v prostředí Revitu

Legendy+ je v uživatelském prostředí Revitu prezentován jako karta "Revit Legends+" v panelu nástrojů "Doplňky" (v anglické verzi Add-ins). Tato karta se chová jako všechny ostatní paletky nástrojů Revitu a lze s ní proto libovolně manipulovat, začleňovat ji do uživatelských pásů karet, nebo z ní udělat plovoucí součást kreslicí plochy.

Karta Legendy+ obsahuje celkem čtyři příkazy, které může uživatel využívat. Jednotlivé příkazy jsou na sobě vzájemně nezávislé a provádějí jasně danou uzavřenou operaci. Tyto operace na sebe logicky navazují a umožňují tvorbu kompletní tabulky s legendou.

Použití jednotlivých příkazů většinou předchází vytvoření vzorové buňky budoucí tabulky (legendy) uživatelem. Takováto buňka může mít libovolný tvar (nejčastěji jeden sloupec, nebo řádek) a obsahuje libovolné množství schémat a textových poznámek,



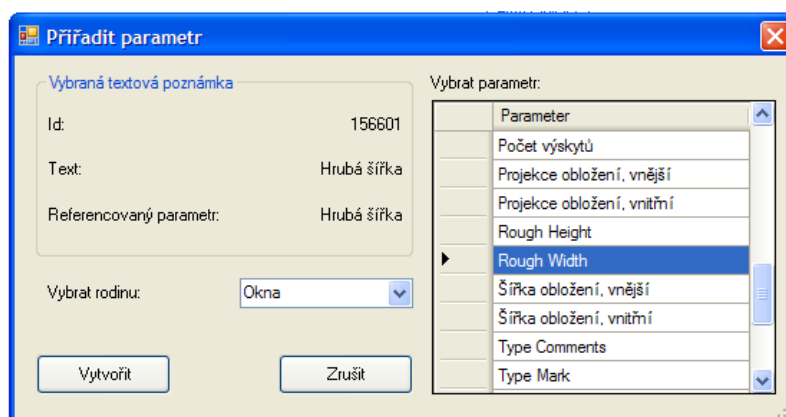
Obrázek 6: Ribbon Legendy+ v panelu doplňků

které mají sloužit pro výpis požadovaných parametrů z projektu. Nad tímto "předpisem" legendy jsou následně používány jednotlivé funkce nadstavby.

4.3.1 Připojit atribut

Příkaz Připojit atribut je k dispozici tehdy, má-li uživatel v aktivním výběru objekty textových poznámek a umožňuje jednotlivým poznámkám přiřadit propojení na reálný parametr libovolné rodiny z modelu. Výsledkem použití tohoto příkazu jsou textové poznámky, které se při budoucím automatickém generování legendy přepíší parametry, které mají reflektovat.

Po spuštění příkazu se uživateli zobrazí formulář s údaji aktuálně vybrané textové poznámky a s nabídkou všech podporovaných modelových rodin, jejichž parametry lze textové poznámce přiřadit. Tyto výpisy jsou generovány dynamicky z aktuálního modelu a dokáží proto zobrazit libovolný parametr, ať již je tento parametr původní součástí Revitu nebo uživatelsky vytvořeným doplňkem. Zároveň to znamená, že jsou podporovány parametry v jakékoliv jazykové verzi Revitu. Pokud má uživatel v aktivním výběru více objektů textových poznámek, příkaz cyklicky projede a stejným způsobem zpracuje každou z nich.



Obrázek 7: Příkaz Připojit atribut

K propojení textové poznámky s modelovým parametrem je v implementaci využito tzv. „sdílených parametrů“. To jsou uživatelsky vytvořené parametry, jejichž definice Revit po skupinách ukládá do volitelných textových souborů. Legendy+ pro tento účel vytváří a editují svůj vlastní soubor se sdílenými parametry (to proto, aby se předešlo konfliktům s jakýmkoliv jinými definicemi parametrů, které může uživatel ve svém projektu využívat).

Tento soubor se nazývá „*RevitLegendsParam.txt*“ a je uložen v kořenovém adresáři instalace pluginu. Ačkoliv obsahuje pouze jednoduché textové označení a definice parametrů, neměl by být nikdy napřímo upravován a vždy by k němu mělo být přistupováno pouze z prostředí Revitu. Legendy+ v tomto souboru vytváří definici instančního textového parametru (tj. takového, který může nabývat různých hodnot pro každou instanci rodiny) „**Legend.Parameter.Reference**“. Tento parametr je v samotném prostředí Revitu pro uživatele neviditelný (a tedy se neobjevuje ve výpisech, nebo exportech) a je ovlivňován pouze v rámci nadstavby. Součástí nadstavby je ošetření veškerých procesů souvisejících s připojením definic tohoto parametru a po ukončení práce opětovné vrácení původních definic a uživatelských nastavení sdílených parametrů.

```
private bool setNewParameterToInstanceTextBinding(UIApplication app, DefinitionFile
myDefinitionFile)
{
    DefinitionGroups revitLegendGroups = myDefinitionFile.Groups;
    revitLegendGroup = revitLegendGroups.Create("RevitLegendParameters");

    parameterReference = revitLegendGroup.Definitions.Create("Legend.Parameter.Reference",
        ParameterType.Text, false);

    CategorySet myCategories = app.Application.Create.NewCategorySet();
    Category myCategory = app.ActiveUIDocument.Document.Settings.Categories.GetItem(
        BuiltInCategory.OST_TextNotes);
    myCategories.Insert(myCategory);

    InstanceBinding instanceBinding = app.Application.Create.NewInstanceBinding(myCategories
    );

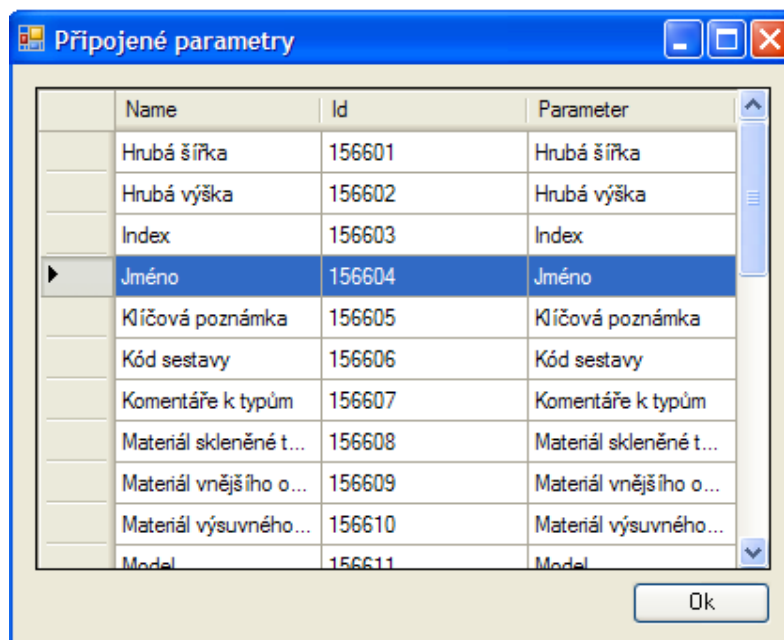
    BindingMap bindingMap = app.ActiveUIDocument.Document.ParameterBindings;
    bool instanceBindOK = bindingMap.Insert(parameterReference,
        instanceBinding, BuiltInParameterGroup.PG_TEXT);
    return instanceBindOK;
}
```

Výpis 5: Algoritmus tvorby a připojení sdílených parametrů

4.3.2 Zkontrolovat text

Příkaz Zkontrolovat text funguje také pouze nad označenými textovými poznámkami. Jeho účelem je poskytnout uživateli rychlou a intuitivní možnost rychle si zkontrolovat propojení textových poznámek v pohledu pro legendy s parametry modelu. Na rozdíl od ostatních příkazů neumožňuje Zkontrolovat text cokoli měnit nebo nastavovat a slouží

pouze ke kontrole jinak neviditelného parametru "Legend_Parameter_Reference" (viz. kapitola 4.3.1).



Obrázek 8: Okno příkazu Zkontrolovat text

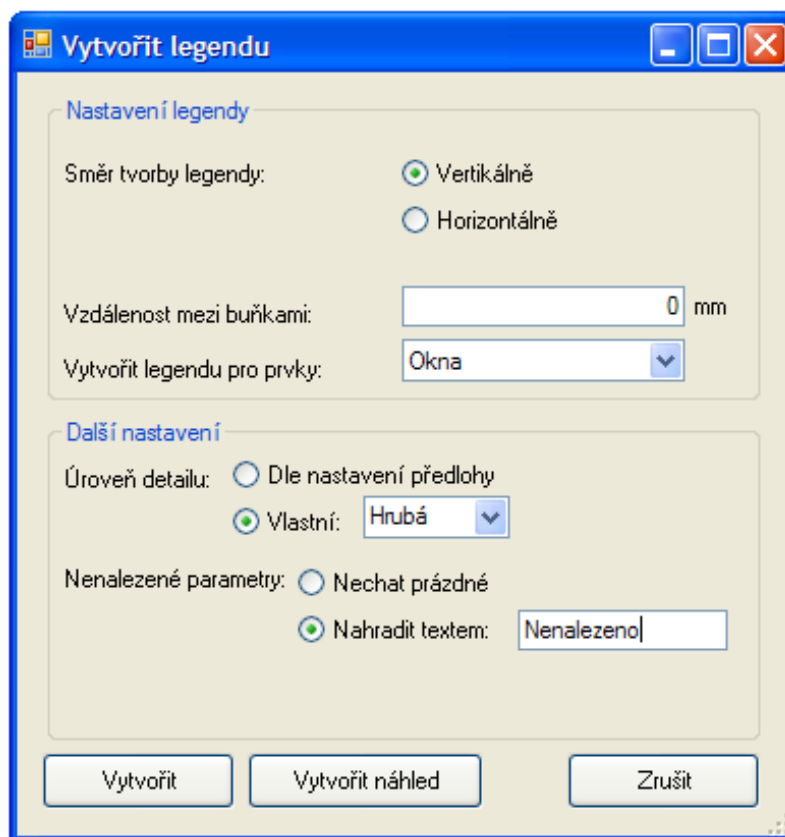
4.3.3 Odpojit atribut

Tento příkaz umožňuje zrušit propojení mezi vybranými textovými poznámkami a jim přiřazenými modelovými parametry. Tímto způsobem může uživatel "vrátit" textové poznámky do původního stavu a nemusí se bát, že by se mu tyto poznámky chovaly nestandardně při budoucí manipulaci s nimi.

4.3.4 Vytvořit legendu

Vytvořit legendu je hlavním příkazem celé nadstavby. Pro jeho úspěšné provedení je nutné aby měl uživatel v aktivním výběru "předpis" pro svou budoucí tabulku s legendou. Takový předpis může být jedním řádkem, nebo sloupcem tabulky. Kromě čar ohraničujících jeho rozměry obsahuje libovolný počet schémat, textových poznámek s referencí na parametry modelu a případně další "holé" texty (například pro nadpisy sloupců tabulky).

Tento základní předpis je následně příkazem rozkopírován tolikrát, kolik se v modelu nachází **typů** uživatelem zvolené **rodiny**. Vzniká tabulka textových poznámek a identických schémat objektů. Texty se v další fázi běhu příkazu přepíší na hodnoty parametrů, které mají odkazovat a schématům se přiřadí odkaz na odpovídající prvky modelu (sché-



Obrázek 9: Příkaz Vytvořit legendu

mata obsahují parametr **ElementID**, který odpovídá ID objektu, jež zobrazují). Vzniká finální podoba legendy.

Grafické okno příkazu nabízí před jeho spuštěním možnost dále jeho běh přizpůsobit požadavkům uživatele. Obsahuje dvě sekce - část „*Nastavení legendy*“, která nabízí nastavení nezbytná pro úspěšné provedení příkazu a část „*Další nastavení*“, bez nichž se příkaz obejde a která uživateli dává větší kontrolu nad výstupní tabulkou.

Prvním mezi nezbytnými nastaveními je parametr **Směr tvorby legendy** s volbami *vertikální* a *horizontální*. Tento parametr je přítomen z důvodu obrovské variability typů a podob legend použitých v praxi. Užívané normy pro legendy se liší v závislosti na prostředí, typu objektů, pro které legenda vzniká a typu projektu do kterého je zahrnuta. Kromě volitelné geometrie legendy je proto nutné uživateli umožnit zvolit i směr její tvorby.

Dále je k dispozici sebevysvětlující atribut **Vzdálenost mezi buňkami**. Standardně je tato hodnota nastavena na 0 milimetrů, což znamená, že jednotlivé buňky legendy na sebe budou bez jakékoliv mezery navazovat okraji své geometrie. Jak jsme si již vysvětlili v kapitole 4.2, jádro Revitu používá pro veškeré délkové výpočty stopy a palce a při



Cena	1006.00	1002.00
Číslo OmniClass	23.30.20.17.21.14	23.30.20.17.21.14
Hrubá šířka	0.0	0.0
Hrubá výška	0.0	0.0
Index	5	6
Jméno	0915 x 1220 mm	0915 x 1525 mm
Klíčová poznámka	08520	08500
Kód sestavy	C	B2020100
Komentáře k typům	komentář okna 6	komentář okna 2
Materiál skleněné tabule	Sklo	Sklo
Materiál vnějšího obložení	Obložení	Obložení
Materiál vnitřního obložení	Obložení	Obložení
Materiál výsuvného okna	nezadáno	nezadáno
Model	model okna 6	model okna 2
Nadpis OmniClass	Casement Windows	Casement Windows

Obrázek 10: Legenda vygenerovaná nadstavbou

programování nadstaveb je proto potřeba myslet na převod těchto nativních jednotek na jednotky používané v aktuálně otevřeném projektu.

Posledním z povinných nastavení je **Vytvořit legendy pro prvky**, které dává na výběr ze všech podporovaných rodin objektů. Jednotlivé rodiny a kategorie se v Revitu rozdělují na dvě skupiny: *modelové* a *anotační*. Mezi modelové patří veškeré objekty používané v modelu projektované budovy. Anotací jsou pak jednotlivými textovými poznámkami, kótami, nastaveními jednotek, pohledů, výpisů, barev a dalších popisových součástí projektu. Ve vnitřní databázi Revitu ovšem tyto dvě skupiny od sebe nejsou odlišeny a všechny jejich prvky se nacházejí ve stejné kolekci rodin. Z toho důvodu není možné, aby Legendy+ dynamicky nabízely tvorbu legendy pro všechny rodiny objektů použitých v modelu. Namísto toho je nabídka podporovaných rodin omezena na předem vybranou množinu.

Mezi nepovinná nastavení patří **Úroveň detailu** a **Chybějící parametry**. **Úroveň detailu** nabízí možnost hromadně nastavit parametr *úroveň detailu* všem schémátům budoucí legendy. Tato úroveň detailu vyjadřuje, co vše bude v rámci schémat viditelné (důležité to může být například u kompozitních stěn skládajících se z více základních druhů, které mohou být důkladně zobrazeny pouze při velké úrovni detailu). Zároveň je možné přednastavit tento parametr zvlášť každému schématu v předpisu legendy a nechat nadstavbu ať operuje s těmito rozdílnými nastaveními. **Chybějící parametry** určuje, jak se naloží s těmi textovými poli, které referencují prázdný nebo neexistující parametr (protože ne všechny objekty té které rodiny musí sdílet všechny parametry). Uživatel

může nastavit, zda takováto pole smazat, nechat prázdná, nebo nahradit vlastním textem (např. „nenalezen“).

Po nastavení všech povinných i volitelných položek následuje generování legendy. Tomu může, ale nemusí (v závislosti na volbě uživatele) předcházet vygenerování náhledu. Tento náhled je jednoduchou tabulkou, která obsahuje všechny prvky vybrané rodiny, které budou obsaženy v legendě. Tímto způsobem si uživatel může zkontrolovat, zda generuje legendu pro správné prvky a zda základní data těchto prvků odpovídají očekávané skutečnosti. To vše ještě před tím, než dojde k časově náročnějšímu generování legendy.

4.4 Testování

Aplikace Revit Architecture má řadu jazykových a vývojových verzí a prostředí ve kterém je využívána jsou různorodá. Z toho důvodu bylo nutné nadstavbu Legendy+ důkladně otestovat ve větším množství variací těchto prostředí a zajistit její funkčnost v každém z nich. Jediná konstanta, která se mezi těmito prostředími nachází je operační systém, jež musí vždy být nějakou verzí Microsoft Windows (to je dáno samotným Revitem).

Pro testování aplikace byly využity následující varianty nastavení aplikace a operačního systému:

- Operační systémy Windows XP, Windows Vista a Windows 7.
- Revit Architecture verze 2011, včetně i bez nainstalovaných dodatečných oprav a service packů.
- Revit Architecture 2011 v české a anglické verzi.
- Revit Architecture bez i s dalšími často používanými volitelnými doplňky a nadstavbami.

Testování probíhalo na reálných modelech budov nacházejících se v databázi ukázkových projektů Revitu Architecture 2011 a na modelech využívaných společností Adeon CZ, s.r.o. pro školení uživatelů Revitu. V rámci testování byly vytvořeny tabulky s rozličnou geometrií ke všem podporovaným typům rodin. Doba nutná pro běh aplikace nepřekročila ani v případě extrémně rozsáhlých modelů řád několika desítek vteřin.

V rámci testování byl odhalen zásadní problém, ke kterému docházelo při spouštění nadstavby v rozdílných jazykových verzích. Důvodem k němu byl fakt, že na lokalizaci Revitu se během poslední dekády podílelo větší množství překladatelských a IT společností. Každá z nich k systému překladač přistupovala jiným způsobem a výsledkem je stav, kdy vedle sebe v databázi Revitu existují názvy parametrů a objektů v lokalizovaném (českém) i originálním anglickém znění. To by samo o sobě nebylo kritické, nicméně často dochází k situacím, kdy Revit přečte libovolný parametr a uloží si ukazatel na něj v jedné jazykové podobě a následně při pozdějším načítání parametru ho dokáže najít jen v odlišné jazykové podobě.

Tento problém se však nakonec podařilo vyřešit kombinací použití unikátních identifikačních čísel některých atributů, chytrým propojením atributů dotazovacím jazykem

LINQ a napevno programátorsky definovaných jmen parametrů. Řešení to není ideální, nicméně je navrženo a implementováno v rámci možností a omezení, které jsou Revitem 2011 nabízeny a v současné době je nejlepším funkčním řešením tohoto problému.

5 Závěr

Tato práce byla vypracována s cílem poskytnout základní popis technologie založené na propojeném databázovém modelu budovy, která je užívána ve stále rostoucím množství CAD aplikací pro oblast stavitelství a infrastrukturu. Byly vysvětleny odlišnosti od jiných přístupů k počítačem asistovanému kreslení a představeny programy, které této technologii s úspěchem využívají.

V druhé části práce je poté podrobněji rozebrána rodina aplikací Autodesk Revit, její specifika a struktura. Součástí tohoto rozboru je představení aplikačního programovacího rozhraní, které si klade za cíl umožnit zájemcům o programování aplikací pro tuto skupinu programů rychle získat znalost tvorby jednoduchých nadstaveb.

Posledním a zdaleka nejdůležitějším úkolem bylo vyplnit jednu ze slabin aplikace Revit Architecture 2011 a umožnit architektům a projektantům pracujícím v českém prostředí nenáročnou tvorbu dynamických legend splňujících zde používané normy. Toho bylo dosaženo nadstavbou Legendy+ využívající Revit API pro automatizaci velkého množství jinak manuálních operací nad pohledem pro legendy a modelem projektu. Tímto způsobem se několikahodinová práce stává otázkou několika desítek vteřin a uživatelé získávají možnost své legendy libovolně a bez další práce aktualizovat a doplňovat o změny provedené ve svém projektu.

Zadání bakalářské práce bylo splněno. Nadstavba Legendy+ byla otestována na pestré škále uživatelských stanic s rozdílnými jazykovými a vývojovými verzemi Revitu. Nadstavba je nyní v plně funkčním stavu šířena mezi uživatele Revitu Architecture 2011. První odezvy na užitečnost nadstavby jsou velmi pozitivní a od uživatelů se objevuje řada návrhů na její další rozšiřování.

Do budoucna by proto bylo vhodné na tuto práci navázat například rozšířením nadstavby o možnost vytvářet legendu i pro instanční prvky rodnin, nebo doplnit legendu o vlastní geometrické náčrty složitějších rodnin, jakými jsou například místnosti. To již předpokládá hlubší znalost rozhraní Revit API a vytvoření algoritmů pro kreslení jednoduchých grafických objektů.

Z hlediska následného vývoje kódu by bylo vhodné shrnout některá lokální nastavení jednotlivých formulářů a příkazů a najít v rámci Revit API (které je v tomto ohledu velmi omezující) způsob, jak tyto uchovávat *globálně* na jednom místě. To by mělo v případě úspěšné implementace za následek zjednodušení kódu a zároveň přineslo komfortnější řešení uživatelům nadstavby.

To všechno jsou, spolu se zajištěním kompatibility nadstavby pro verzi Revit Architecture 2012 (která vyjde v půlce roku 2011) cíle budoucího vývoje pluginu Legendy+.

6 Literatura

- [1] SUTHERLAND, Ivan Edward, preface by Alan Blackwell and Kerry Roddenphone (September 2003). *Sketchpad: A man-machine graphical communication system* [online]. Technical Report No. 574, University of Cambridge. Dostupné z WWW: <<http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-574.pdf>>. UCAM-CL-TR-574.
- [2] *Autodesk.com* [online]. 2011 [cit. 2011-04-09]. Dostupné z WWW: <www.autodesk.com>.
- [3] EASTMAN, Charles M. *BIM Resources @ Georgia Tech* [online]. 2009 [cit. 2011-01-04]. Building Information Modeling. Dostupné z WWW: <<http://bim.arch.gatech.edu/?id=402>>.
- [4] YESSIOS, Chris I. Are We Forgetting Design? *AECbytes Viewpoint* [online], 2004, 10 [cit. 2011-01-04]. Dostupné z WWW: <http://www.aecbytes.com/viewpoint/2004/issue_10.html>.
- [5] EASTMAN, Chuck. *BIM handbook : a guide to building information modeling for owners, managers, designers, engineers, and contractors*. 2008. 490 s. ISBN 978-0-470-18528-5.
- [6] Industry Foundation Classes. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 9 November 2004, last modified on 1 April 2011 [cit. 2011-04-10]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Industry_Foundation_Classes>.
- [7] *ArchicadWiki* [online]. c2006 [cit. 2011-04-10]. Dostupné z WWW: <<http://www.archicadwiki.com>>.
- [8] *Bentley Systems, Incorporated* [online]. c2011 [cit. 2011-04-10]. Dostupné z WWW: <<http://www.bentley.com>>.
- [9] GREENWOLD, Simon. *Autodesk.com* [online]. 2004, 2005-2007 edits by David Driver [cit. 2011-04-08]. Building Information Modeling with Revit Architecture. Dostupné z WWW: <http://images.autodesk.com/adsk/files/rac_2008_curriculalecturenotes_final.pdf>.
- [10] *Revit 2011 API Developer Guide* [online]. 2011 [cit. 2011-04-09]. Dostupné z WWW: <<http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=2484975>>.